## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A disk scheduling system, comprising:

at least one queue to hold a subset of Input/Output requests directed to a storage medium; and

a scheduling component that employs a predetermined number of the requests within a round of scanning the storage medium to provide a ~~particular~~ maximum latency ~~level~~ period for servicing the requests within the round, the maximum latency period is determined based on number of requests within the round, and reorders the requests to optimize scheduling of the requests to maintain ~~a particular~~ an optimal throughput level in connection with storage medium updates.

2. (Currently amended) The system of claim 1, the scheduling component is associated with a categorizer that automatically directs the requests to one or more classes of queues.

3. (Original) The system of claim 2, the scheduling component includes a queue of predetermined size that receives the requests from the categorizer.

4. (Original) The system of claim 1, the storage medium includes at least one of a hard disk, floppy disk, memory stick, compact disk, and a memory that is scheduled to be accessed from a computer.

5. (Original) The system of claim 1, the scheduling component dynamically calculates a determined amount of bounded latency in order to perform I/O scheduling for selected tasks.

6. (Original) The system of claim 1, further comprising at least one of a periodic queue and an aperiodic queue.

7. (Currently amended) The system of claim 6, the periodic queue is arranged in an Earliest-Deadline-[[f]]First ordering.

8. (Original) The system of claim 6, the aperiodic queue is arranged according to criticality classes.

9. (Original) The system of claim 8, the criticality classes include at least one of critical, high, interactive, normal, background, low, and idle.

10. (Original) The system of claim 8, the aperiodic queue is arranged according to a First-In-First-Out (FIFO) ordering.

11. (Original) The system of claim 6, further comprising a sweep queue to process the periodic queue and the aperiodic queue and to access the storage medium.

12. (Currently amended) The system of claim 11, the sweep queue is managed according to a C-LOOK component and the requests in the sweep queue are arranged in C-LOOK order.

13. (Original) The system of claim 6, the periodic queue is associated with periodic I/O associated with multimedia applications.

14. (Original) The system of claim 6, further comprising a component to process periodic I/O streams having performance parameters that are checked in view of system resources.

15. (Original) The system of claim 6, further comprising a component that analyzes a periodic I/O stream in view of system capabilities and aborts a request if the capabilities are exceeded.

16. (Currently amended) The system of claim 15, the component is an admission control component~~. (not shown)~~ and enforces that respective parameters of the periodic I/O are not exceeded.

17. (Original) The system of claim 16, the admission controller includes a deadline component to assign a deadline or timeframe to a request in which the request is to be completed.

18. (Currently amended) The system of claim 1, further comprising the following equation to determine ~~access time~~ the maximum latency period to service a round of requests to update the storage medium:

$$service(n) = n\left(time\_seek\left(\frac{Cylinders}{N}\right) + time_{transfer} + time_{rotation} + time_{controller}\right) + time_{sweep}$$

wherein $n$ is an integer and represents the number of the requests within the round, and *service(n)* represents the maximum latency period for servicing the requests within the round.

19. (Currently amended) A computer-readable medium having computer readable instructions stored thereon for implementing the ~~scheduling component and the queue~~ system of claim 1.

20. (Currently amended) A method to schedule requests to a storage medium, comprising:

determining a predetermined number of requests for a set of requests;

determining a maximum latency period ~~parameter associated with the set of requests~~ for servicing the set of requests to update the storage medium, the maximum latency period is determined based on the number of requests within the set of requests;

reordering the requests to optimize scheduling of the requests to maintain an optimal throughput level; and

updating [[a]] the storage medium ~~based upon~~ by servicing the set of requests within the maximum latency period ~~parameter and a desired throughput for the set of requests~~.

21. (Original) The method of claim 20, further comprising automatically separating Input/Output streams associated with the set of requests into at least one of periodic streams and aperiodic streams.

22. (Currently amended) The method of claim 21, further comprising:

arranging the periodic streams in an Earliest-Deadline-First ordering; and

arranging the aperiodic streams in a First-In-First-Out ordering.

23. (Original) The method of claim 22, further comprising applying a C-LOOK algorithm to the periodic streams and the aperiodic streams to access a storage medium.

24. (Withdrawn) The of claim 20, further comprising applying functionality associated with the following instructions to access the storage medium:

```
while periodic-queue ≠ Ø and periodic-slots ≠ 0 and
    DEADLINE (periodic-queue) < ROUND_TIME do
  INSERT_LBA_ORDER (sweep-queue,
    REMOVE_EDF_ORDER (periodic-queue))
  periodic-slots = periodic-slots - 1
end

for priority = High downto Idle do
   while priority-list [priority].slots ≠ 0
```

```
            INSERT_LBA_ORDER (sweep-queue,
                REMOVE_FIFO_ORDER (priority-queue [priority])
            priority-list [priority].slots = priority-list [priority].slots - 1
        end
    end

    while sweep-queue ≠ Ø do
        SCHEDULE (REMOVE_MINMUM_LBA (sweep-queue))
    end
```

25. (Currently amended) A method to perform disk updates, comprising:

automatically determining a maximum latency ~~requirements~~ period to service ~~for~~ a round of requests to a storage media in order for an application to function;

automatically adjusting ~~a size for~~ the number of requests to be ordered for the round of requests, so the number of requests within the round after any such adjustment is the maximum number of requests that are able to be serviced within the maximum latency period;

ordering the round of requests, so adjusted; and

within the maximum latency period, updating [[a]] the storage media [[with]] by servicing the round of requests.

26. (Currently amended) The method of claim [[27]] 25, further comprising:

determining bandwidth requirements for the round of requests; and

automatically prioritizing the round of requests based upon the determined bandwidth requirements.

27. (Currently amended) The method of claim [[28]] 26, further comprising dynamically monitoring the bandwidth requirements and the maximum latency ~~requirements~~ period.

28. (Currently amended) A system to facilitate disk updates, comprising:

means for classifying a set of requests;

means for queuing the set of requests having a fixed size;

means for interacting with a storage medium based upon ~~at least one of~~ a maximum latency ~~requirement~~ period for servicing the set of requests to update the storage medium, the maximum latency period is determined based on the number of requests within the set, and [[a]] an optimal throughput ~~requirement~~ level that is maintained by reordering the requests to optimize scheduling of the requests; and

within the maximum latency period, servicing the set of requests to update the storage medium.

29. (Currently amended) A computer-readable medium having a data structure stored thereon for aiding in the implementation of the system of claim 1, comprising:

a first data field related to a size parameter for a queue associated with a round of requests; and

a second data field related to a maximum latency ~~parameter~~ period for servicing a round of requests to update a storage medium, ~~associated with~~ the maximum latency period is determined based upon at least one of the round of requests and the size parameter.